

Web Designing: Is a task of designing Web Components for a Website.

Web Page: Is a major component of a website. It is a formatted 'file' which appears in the web browser.

Other components of a Web Site: Database, Multimedia Files (Image, Audio, Video, & Graphics)

Web Browser: Is a 'software' which is being used to view the Websites & their components.

Examples: Google Chrome, Microsoft Internet Explorer, Mozilla Firefox, Safari, Opera...

HTML: referred as 'Hypertext Markup Language', used to **create** the 'Web Pages'.

CSS: referred as 'Cascade Style Sheet', a file used to **improve** the *style & attractiveness* of a web page.

Javascript: It's a scripting Language used to **improve** the *interactivity* of a web page.

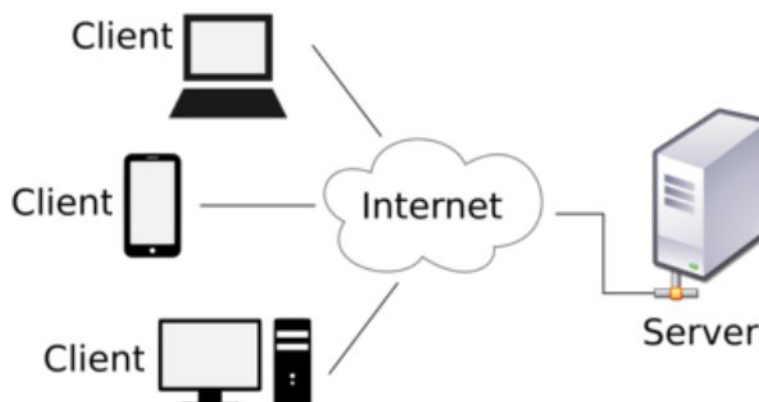
Server: Is a computer connected to the internet and it provides services to the users. The services may be containing a website & it's components (Web Hosting), storing the files of a user (Ex: Google Drive), containing multimedia files (Movies, MP3 Songs, eBooks *etc.*)

Client: Is a computer a user used to get the services from a server in the Internet.

An Example:

If you are viewing '*facebook*' through '*Google Chrome*' in the Internet... then Facebook is a Server, Google Chrome is a Browser, & your computer is a 'Client'. Your **Smartphone** also may be a Client.

Server-Client Architecture



Actually what is happening? When you type the address of a website (www.facebook.com) in the address bar of your web browser, then your web browser will send a 'request' to the 'Server' of *facebook* as 'Please show me the facebook's Website'. Then the server receives your request & shows the facebook's web site into your web browser.


How a Website is functioning..?

‘At least one’ server is needed to bear a web site. As stated before, actually a collection of Web Pages & some other components are forming a Website. The web pages are a particular type of ‘*files*’ which reside in a web server & occupy a ‘*web space*’. Hence, the other components such as Multimedia files & databases are usually quite larger in size than web pages then they occupy more ‘web space’ in servers.

Let’s create a Webpage..!

We have to make at least one web page for any website, otherwise that website is just a space & useless. To create a web page, we suppose to write something using a language called ‘HTML’ in any ‘*Text Editor*’ which is software used for creation of ‘text files’. Then we have to save that text file as a ‘Web Page’ file. Usually the text editors we are using are ‘Notepad’, ‘WordPad’ & several others too.

Practical 1: Write our first web page...

1. Open the software ‘Notepad’ by looking this icon  in your computer.
2. Type inside the notepad as following...

```
<html>
<head>
<title>www.MyWebsite.com</title>
</head>
<body>
Hello Everyone... This is my website...!
</body>
</html>
```

3. Now, save the file with the name of ‘**index.html**’. Click the file menu on the top left corner & click ‘Save’ option; then give the name as stated above.
 4. Now, go where you saved that file. Find it and double-click it. Observe what is happening...
-

Simply html is a way of writing a web page. The contents of a web page would be demarcated with *Tags*. Every tag is enclosed with < > symbols. Every html file (webpage file) begins with <html> opening tag & ends with </html> closing tag. Likewise, each component may have both opening & closing tags; but not at all!

Contents of a webpage resemble their expressions on the Web Browser. For example, in the above ‘code’ the phrase inside the ‘title’ tags (www.MyWebsite.com) would appear on the Title bar or the Tab of the Web Browser. Remember, the title tags have to be placed inside the <head> </head> tags, where anything inside of them would not appear inside the Browser. Unless the ‘title’ anything would be placed inside the <body> </body> tags. (Hello Everyone... This is my website...!)

Now let’s do some formatting on the text of a web page as we already known, ‘bold’, ‘italic’, ‘underline’, ‘center’ with the following example. Try the following code just above the </body> tag. Save the file and ‘refresh’ the page which has been already opened in the Web Browser.

This is Bold

<i>This is italic</i>

<u>This is underlined</u>

<center>This is centered</center>

If you want to break the text in to the next line, put this tag **</br>** which has no opening tag..!

We can specify the appearance (or behavior) of a text by stating one or more *attributes* for a tag. Try...

<p align="justify">

Julian Paul Assange (born 3 July 1971) is an Australian computer programmer, publisher and journalist. He is editor-in-chief of WikiLeaks, an organisation which he founded in 2006. He currently resides in Ecuador's London embassy after having been granted political asylum in August 2012.

</p>

Then change the tag **<p align="justify">** as **<p>**. Save it & refresh the page again. You can see that initially the paragraph (denoted as **<p>** tag) was in a 'justified' format but later it became 'left alignment'. Thus we specialized the 'Alignment' for a Paragraph via an Attribute (here it is **align**) & put a 'value' for it (here it is **justify**).

Exercise: Change the values of 'align' attribute as following...

- right
- left
- center

Save the webpage & refresh it. How the paragraph has changed regarding each of the given values? You may notice that without any alignment attribute, the paragraph takes 'left alignment' as the default.

Add an image in to a webpage can be done with a tag which has no 'Opening Tag', as shown below.

Here the **** tag takes three attributes as 'src', 'height' & 'width'. It is noticeable that the attribute 'src' is essential for the image's adding. The value of that 'src' attribute is actually the "source" of the image in your computer. 'Actually where that image is in..?' If the attributes 'height' & 'width' aren't mentioned, the image will be displayed with its original dimensions.

Every image has its file extension which tells the type of the image. There are three or four popular image formats are being used; **JPG, PNG, BMP, & GIF**. Any image type can be included through the **** tag. In the above example the image type is 'JPG'.

It also can be noticed that **** tag also wouldn't have the pair of Open & Close tags.

If you need to put a *horizontal straight-line*, the pair of tags are **<hr>** & **</hr>**. Try it between two paragraphs and or images.

Headings

There are six sizes can be used for 'headings' of a text, in a web page. The code below shows the biggest & smallest sizes of headings. Hence we can notice that once after the closing tag, the text would be broken to the next line. Replace the code with <h2>, <h3>, <h4>, <h5>, <h6> & see the result.

<h1>Bigger Heading</h1>

<h6>Smaller Heading</h6>

Font, Font-Size & Color of a text

We can change the font, font-size & color of a text by using the tag which may have attributes 'face', 'size', and 'color'. Try the following codes & see the differences among them.

**Any Difference?
</br>**

Any Difference?

Table

We may have seen Tables in any text-documents, that probably have rows & columns with may or may not be a 'Header'. Unlike other HTML tags, to make a table we have to use at least three pairs of tags as <table></table>, <tr></tr>, <td></td>. Below example shows a table with 3 rows & 2 columns. In addition, there is an attribute 'border' which determines the thickness of table-border.

```
<table border="1">  
<tr> <td>Cell1</td> <td>Cell2</td> </tr>  
<tr> <td>Cell3</td> <td>Cell4</td> </tr>  
<tr> <td>Cell5</td> <td>Cell6</td> </tr>  
</table>
```

Here, <tr> tag is used for define a 'row' of a table & inside of that, <td> tag is used for 'Table Data'. We can put a header in this table as just replace the <td> tags as <th> in the first row only. The 'align' attribute with the <td> tag can align its contents as 'left', 'right', 'center', & 'justify'. However the <th> tag has already appeared as both 'center-aligned' & 'Bolded Text'.

colspan: In a table, there may be number of columns are merged in a certain row, whereas other rows are remaining same. This feature is called as 'colspan', which is an attribute of <td> tag & it specifies how many columns would be merged. Try the example given below.

MergedCell	
Cell 3	Cell 4

```
<table border="1">  
<tr> <td colspan="2">MergedCell</td> </tr>  
<tr> <td>Cell 3</td> <td>Cell 4</td> </tr>  
</table>
```

rowspan: In a table, there may be number of rows are merged in a certain column, whereas other columns are remaining same. This feature is called as 'rowspan', which also is an attribute of <td> tag & it specifies how many rows would be merged. Try the example given below.

MergedCell	Cell 3
	Cell 4

```
<table border="1">  
<tr> <td rowspan="2">MergedCell</td> <td>Cell 3</td> </tr>  
<tr> <td>Cell 4</td> </tr>  
</table>
```

Although the <table> tag was very useful to place the various portions in a webpage, those web-pages are noticed with a less-responsiveness with the browsing devices such as Monitors with various screen-sizes, Mobile phones, Laptops etc. Therefore, in order to make a responsive web page, nowadays the <div> tag is being used popularly. Especially the <div> tag is most convenient to use with 'CSS' scripts.

Lists – Ordered & Unordered

To show an ordered list & its list-items, there are two pairs of tags as ``, `` & `` & `` respectively. It begins with `` tag to start the list & each of list item can be enclosed with `` & `` tags, finally we can finish the list with `` tag. A list can be specialized by 'type' attribute which may have the values as '1', 'a' or 'A' or 'i' or 'I'. Try this ordered list, uses roman letters to number its items. An unordered list begins takes `` & `` tags instead of `` & `` tags, and takes style attribute (CSS) through which it can be specialized, as following: `<ul style="list-style-type:disc">` and replace 'disc' with *circle*, *square*, *none*.

```
<ol type="i">
<li> Baby </li>
<li> Child </li>
<li> Youth </li>
<li> Elder </li>
</ol>
```

Detailed List..?

It is another type of list. Apply the following code and see the output. How this detailed list is differed from both ordered & unordered lists?

```
<dl> <dt>Coffee</dt> <dd>Black Coffee</dd> <dt>Milk</dt> <dd>White Milk</dd> </dl>
```

Hyperlink

A webpage has an ability to link another webpage itself, with a feature called "Hyperlink". Assume we have two web pages namely 'page1.html' & 'page2.html'. In order to link these pages mutually, both pages have to have a link which linked with other page. To link another page, user may have to do something (usually 'clicking' by mouse). Thus, we assume a text phrase 'Click Here' in 'page1.html' has the link to 'page2.html', then the code suppose to be... `Click Here`

The `<a>` tag is 'Anchor' tag has an attribute 'href' (host reference). You might see the linked webpage 'page2.html' will be opened in the 'same tab' where you were in 'page1.html'. Try to put an attribute & its value as `target="blank"`. Now the page2.html will be opened in a 'different tab' on your browser. Now how can you go back to the previous page 'page1.html'?

Inputs

Initially web-pages were used for just deliver something, content / material / service to the users. Later they became more interactive and ability to get something from users & behave accordingly. User can give inputs and server will response them via a webpage. These inputs can be taken via various ways depending on the type & nature of an input. Let's assume a user gives his/her details to 'Sign Up' in to an account (Ex. Facebook). What are the details he/she wants to give? Name, Gender, Email Address, Date of Birth, Country, Phone Number etc, of course these are different types of inputs. 'HTML 5' is the version which supports these inputs very well.

`<input>` tag is used (no closing tag) to get inputs from users, & it is specialized by its attribute 'type' which takes values as 'text', 'number', 'password', 'date', 'time', 'month', 'range', 'color', 'week' and so on. Some additional attributes arise as *min* & *max* for the values number, date time, to ensure a range of them. Also an attribute '*name*' is used to differentiate the inputs among them.

Ex: `<input type="date" name="birthdate" min="1916-12-31" max="2015-12-31"/>`

For an easy identification of each input, it is better to put a “Label” for them. Labeling can be done via using the `<Label> </Label>` tags and put an attribute ‘for’ with a value which is exactly the value of the attribute ‘id’ of that particular input. Then, if you click on that label, the curser will go inside the input & ready to enter or do something. Let’s try the following example.

`<label for="username">My Username</label><input type="text" id=" username ">`

Radio-buttons & Check-boxes

These also input types but these differ from earlier inputs because of mouse-selective nature. Both of them are used for “select something from many”. Through ‘radio-buttons’ a user can select only ‘one item’ form several items, whereas through ‘check-boxes’ a user can select an item form several items. Try the code below.

For Radio-buttons...

`<input type="radio" name="choice" value="Right" checked> Correct

<input type="radio" name="choice" value="Wrong"> Incorrect
`

For Check-boxes...

`<input type="checkbox" name="vehicle1" value="Bike"> I have a bike

<input type="checkbox" name="vehicle2" value="Car"> I have a car
`

You may see, in the radio-buttons, the attribute ‘name’ has the same value (‘choice’) whereas in the checkboxes the attribute ‘name’ has two different values (vehicle1 & vehicle2).

It is noticeable that some values of the ‘type’ attributes are no longer supported by some browsers.

Ex: URL, DateTime, Email, Tel.

Buttons

Buttons (clicking) also are used to give some inputs to Server, vi a webpage. There are 2 ways to make a button. One is via `<input>` tag & another is `<button>` tag. However both of them are apparently similar.

`<input type="button" value="Click Me"/> <button>Click Me</button>`

Special Types of Buttons

A button may be pre-determined for its behavior as ‘submit’ or ‘reset’, where submit is submitting the entered inputs and reset is clearing all inputs given in a ‘form’. (A ‘form’ apparently has several (different types of) inputs and carries the data from a user to a server. Try the following codes.

`<input type="submit" value="O.K"/> <input type="reset" value="Clear"/>`

We’ll study more about ‘buttons’ under ‘Javascript’ where buttons are very useful to do several things.

Tags for Multimedia Elements

It is obvious that we can see the multimedia components such as video, audio, graphics etc., in any webpage nowadays. Once a multimedia file; a MP4 video file has added in the server, we can embed that video similarly how we added images previously. Look the code below.

```
<video width="320" height="240" controls> <source src="movie.mp4" type="video/mp4">
</video>
```

Here, the <video> tag has a ‘sub tag’ named ‘source’ (without the closing tag) which specify which video file has to be embedded & what is the format of that video. Also you may notice an attribute ‘controls’ doesn’t take any values but it causes the appearing of Video Controls such as ‘Play-Pause Buttons’, ‘Volume Up/Down’ & ‘Fullscreen’ etc. Without putting this ‘control’ attribute, the video cannot be played on a webpage.

If you need to embed an audio file; An ‘mp3 file’ you can use audio tag which behaves similar to video tag as shown below. Here the height & width attributes aren’t applicable.

```
<audio controls><source src="song.mp3" type="audio/mp3"> </audio>
```

The above tags are particularly assigned for Video & audio embedding. Additionally, there is another tag by which we can add whatever the multimedia file format, even some text files too (PDF). The <object> tag is widely used nowadays to embedding a content of a web page in to another site’s webpage.

```
<object width="400" height="400" data="document.pdf"></object>
```

We can simply use this object tag to embed a video or an audio file. However the difference is, when we use this tag, once the webpage is loaded, the audio or video would play automatically..! Check it...!

Comments in HTML

In any programming script, the ‘Commenting’ takes an important role. A comment in a programming script is simply expressed as “*Appears in input but doesn’t appear in output*”. Look the code below.

```
<html>
<body>
Students are coming
<!--Teachers are waiting-->
</body>
</html>
```

In the above, you can see the text “Teachers are waiting” doesn’t appear on the output webpage. This means, that ‘text’ *has been commented*, although it is included in the input. This behavior can be used for the following reasons.

1. An explanation about the program-script to the non-programming person (Eg. Manager)
2. Easy to implement some scripts which may change often within the development.
3. To "hide" a script from browsers which aren’t support to that scripts (so they don't show it as text)

Cascade Style Sheet (CSS)

Up to now we practiced how to apply some basic expressions in to a web page. Hereafter we will focus how to enhance the *style* of a webpage. Before that we have to look an important tag called ‘<div>’ tag.

The <div> tag defines a *division* or a *section* in a webpage. The <div> tag is used to group block-elements to format them with CSS. <div> tag takes several attributes to enhance several features of a particular division in the webpage. In fact, without using any attributes the <div> tag is useless.

`<div align="center">This is Center</div>` vs `<center>This is Center</center>`

It is pretty sure that the outputs of the above codes are exactly similar. However div tag can be used to differentiate several sections (or areas) in a webpage & enhance the ‘style’ of them via CSS scripts. Try the following code and see the ‘style enhanced text’.

`<div align="center" style="color:blue; background:orange; border-radius: 15px; width:20%">`

Here the ‘style’ attribute contains several specifications. Those are seem to be ‘color’, ‘background’, ‘border-radius’, and ‘width’, & called as ‘*properties*’ moreover they are having values as well. Actually the entire data inside the ‘style’ tag is called as “Cascade Style Sheet Script”, thus according to this script the behavior of that ‘<div>’ tag differs. Now replace the ‘div’ tag as ‘h1’ tag and see the output.

Therefore, CSS can be understood as ‘*a script, that can be embedded or linked or inserted in to a HTML code, thereafter one or more HTML tags (not only the <div> tag) would appear according to the specifications given under that script*’.

CSS is primarily used with HTML for the ‘Style Enhancement’ and it can be applied as following ways.

1. Styling a single HTML element (Ex: a single <div> Tag)
2. Styling similar tags at once (Ex: All <div> Tags in a Webpage)
3. Styling similar tags but differently (Ex: One <div> Tag is differed from another <div> tag)
4. Styling several Webpages (in a Website) at once (Ex: Every page in **Facebook** has a blue banner)

In addition, CSS can be used for...

1. Change the style-behavior of normal HTML tags (Ex: A ‘List’ can be appeared as ‘Menu Bar’)
2. Create some user-defined HTML tags & define their style-behavior (Ex: New Tag in your Name)

Styling a single HTML element

Here, an attribute ‘style’ for any appropriate HTML tag, to change it’s style. Look the code below.

`<h1 style="color:red">REDTEXT</h1>` Here the heading’s color is changed as ‘Red’. Here the <h1> tag has the ‘style’ attribute & that attribute has 2 regions as ‘**Property**’ (color) and a ‘**Value**’ (red).

Now check this code: **<h1 style="color:red; background:green">REDTEXT</h1>**

Here the 'style attribute' has two properties and two values. Likewise the number of properties and their values may be infinite for the 'style' attribute of particular HTML tag. Check this behavior for these tags and observe the output. , <i>, <u>, <p>

This way of applying is called as '*Inline CSS Styling*'. It is appropriate for a single or few HTML-Tags.

Styling similar (homogenous) tags at once

Obviously a web page having several html tags, then it is difficult to styling each & every tag. Then we can use a trick 'Internal CSS', where we use <style> </style> tags **inside the <head> </head>** tags, and inside of <style></style> tags we are going to specify the CSS script as following.

```
<html><head>
<style>
div{
color:blue;
background:orange;
border-radius: 15px;
width:20%;
}
</style>
</head><body>  <div>First Div</div> </br> <div>Second Div</div>  </body></html>
```

Here, all '<div> tags' in this webpage will be appeared as their style has been decided inside the <style></style> tags. Similarly whatever the HTML element (tag) in this webpage can be styled using this 'internal' CSS implementation. However if we use both 'internal' and 'inline' together the browser will give the priority for 'inline' styling. As an example, modify the given code as following & observe the output.

<div style="color:red; background:green">Second Div</div>

Styling several Webpages (in a Website) at once

Obviously a website having several webpages, then it is difficult to styling each & every page even via internal CSS implementation. Then we can use a trick 'External CSS', where we use a separate CSS file which contains the CSS Code, and link that CSS file in to several HTML files. Look the code below and make these files in a same folder. Observe the outputs of every HTML file.

style.css

```
div{
color:blue;

background:orange;

border-radius: 15px;

width:40%;
}
```

1stPage.html

```
<html><head>
<link rel="stylesheet" type="text/css" href="style.css">
</head><body>
<div>First Div on First Page</div></br><div>Second Div on First Page</div>
</body></html>
```

2ndPage.html

```
<html><head>
<link rel="stylesheet" type="text/css" href="style.css">
</head><body>
<div>First Div on Second Page</div></br><div>Second Div on Second Page</div>
</body></html>
```

Styling similar tags but differently (Ex: One <div> Tag is differed from another <div> tag)

We can use **Separators** for differentiate a tag from another similar tag. The separators are 'id' & 'class'.

Giving an 'id' for each and every tag and specify the CSS styling for each & every id

```
<html><head><style>
#first{color:green;}
#second{color:red;}
</style></head><body>
<div id="first">This is 1st ID</div>
<div id="second">This is 2nd ID</div>
</body></html>
```

Here, we gave the 'id' attribute to <div> tags and inside of <style></style> tags we specify their CSS styling corresponding to their 'id's as they began with '#' hash.

Make a 'Class' for one or more similar tags and specify the CSS styling for each & every class

```
<html><head><style>
.first{color:green;}
.second{color:red;}
</style></head><body>
<div class="first">This is 1st ID</div>
<div class="second">This is 2nd ID</div>
</body></html>
```

Here, we gave the 'class' attribute to <div> tags and inside of <style></style> tags we specify their CSS styling corresponding to their 'classes' as they began with '.' dot.

Then what is the difference between 'id' & 'class'..? Try the following code...

```
<html><head><style>
.first{color:green;}
.second{background:red;}
</style></head><body>
<div class="first second">This is 1st Div</div>
</body></html>
```

Here this <div> tag is having two different properties (color & background) from two different 'classes'. This 'class separator' has this special ability, where the 'id' separator doesn't enable this feature. Another thing, if we had two similar properties inside of these two classes, then the result is unpredictable.

visibility

We can simply hide/reveal an HTML element by using the 'visibility property as following.

```
<div style="visibility:hidden">Hidden</div>
```

And in order to reveal it again, simply replace the value 'hidden' as 'true'. Similar, but not exact thing can be done with <input> tag, where if you want to get an input from user without his/her attention (somewhat associated with server side script ex: php / jsp / asp), you can use the following code.

```
<input type="hidden" value="IP_ADDRESS"/>
```

Positioning & Padding

In CSS, there are several options for 'positioning' of HTML element. If we take a <div> tag, it can be positioned by using three ways such that 'absolute', 'relative', 'fixed' & 'static'. In the 'static' option, without using these ways we can determine the location of that <div> tag via specify where should it be located by 'distances' from 'Top', 'Right', 'Bottom' & 'Left'. (Actually 'Top' & 'Left' are enough to get a position) Here the distance is called as 'padding'.

Try the following code...

```
<html><head><style>
div{height:50px;width:50px;}
#abs{
position:absolute; left:300px; top:50px; background:violet;
}
#rela{
position:relative; left:50px; top:50px; background:red;
}
#fix{
position:fixed; left:150px; top:50px; background:blue;
}
</style></head><body><div id="rela">Rel</div><div
id="fix">Fix</div><div id="abs">Abs</div></body></html>
```

Here, three <div> tags are mentioned as each of them has different 'position' property. We can see when you scroll the webpage, the 'absolute' & 'relative' 'div' portions are moving, while the 'fixed' portion doesn't.

Moreover, if we measure the actual pixel values for 'left' & 'top' padding of 'relative' portion is somewhat adjusted.

The 'padding' property can be expressed as following.

```
<html><head><style>
div{width:40%;}
#pad{
padding-top:50px;
padding-right:30px;
padding-bottom:50px;
padding-left:80px;
border: dotted;
}
#alone{
padding: 50px 30px 50px 80px;
border: dotted;
}
</style></head><body>
<div id="alone">test</div><div id="pad">test</div>
</body></html>
```

Here the CSS styling preferred under both #pad & #alone 'id's are same, and therefore we would expect that both 'div' portions are going to be overlapped. But it doesn't occur, where the second 'div' portion's 'top padding' would be taken as 'not from the monitor-border' but from the 'previous div portion'. Observe the output clearly.

Another note is, if we used the positioning properties 'top, bottom, left, & right' without using of any positioning properties such 'absolute', 'relative', & 'fixed', those don't work. Check a code in such example.

Therefore, there are several ways to position a HTML element in a webpage, & that way of depends on the dimension of webpage

Applying Appropriate CSS styles for Appropriate HTML Tags

It is noticeable that the html tags don't obey to CSS if they are specified by inappropriate CSS styling. As an example, we can apply the 'Border' property for <div>, <p>, <table>, tags. However it isn't meaning full that adding 'Border' property for
 tag, hence nothing will happen if it is applied.

Moreover, it's not necessary to memorize the 'All Properties & their Values' of CSS styling, rather keep them in a PDF file or a HTML file. Because while designing a webpage, nowadays programmers don't keep them in memory rather simply invoke them via the 'Web Authoring Tools' (Ex: Dreamweaver). The several CSS properties and values (with appropriate examples) can be referred via this URL. <http://www.htmlhelp.com/reference/css/properties.html>

hover property

In CSS style sheet, by using this ‘hover’ property, a HTML element (such as <div>, , <a>, etc) can express ‘two styled appearances’ and those appearances can be switched whenever the mouse cursor is moved over it. This feature helps to make some elements ‘responsive’ in a webpage; however it is still being considered ‘A client side scripting language like Javascript or VBScript’ is needed to make a sophisticatedly responsive webpage. Following code shows some elements with this ‘hover’ property.

```
<html><head><style>

div:hover{background-color: yellow;}
a:hover{background-color: red;}
li:hover{background-color: orange; width:20%;}

</style></head><body>

<div style="height:20%;width:20%">This is Div</div>
<a href="#">This is Link </a>
<ul>  <li>Item 1</li>    <li>Item 2</li>    <li>Item 3</li>    </ul>

</body></html>
```

This hover property is often used in *Navigation Bar*, *Dropdown Menu*, and *Tooltip Boxes*. Hence, the Navigation Bar is often associated with dropdown menus. Below Code shows a simple Tool-tip Box.

```
<style>
.dropdown {
position: relative;
display: inline-block; }

.dropdown-content {
display: none;
position: absolute;
background-color: #f9f9f9;
min-width: 160px;
box-shadow: 0px 8px 16px 0px rgba(0,0,0,0.2);
padding: 12px 16px;
z-index: 1; }

.dropdown:hover.dropdown-content {
display: block; }
</style>

<div class="dropdown">
<span>Mouse over me</span>
<div class="dropdown-content"><p>Hello World!</p>
</div></div>
```

Here, the code has two classes as *dropdown* & *dropdown-content*, where the second class *dropdown-content*’s visibility has been hidden by the *display* property as ‘none’. However when the mouse cursor is moved over the element which has been specified as *dropdown* class, the second class has to be appeared. That behavior is defined as...

```
.dropdown:hover.dropdown-content {
display: block; }
```

Nothing special in this code except the *display* property, which specifies an element’s similarly with the visibility property (hidden/true).

We can use this **hover** property in a sophisticated manner, to create **Menu Bar** which may have several dropdown menus.

Menu Bar

```
<html><head><style>
ul { width:232px;}
li { float: left;}
li a {
    display: block;
    text-align: center;
    padding: 14px 16px; }

.droplink {
    background-color: orange;
    padding: 16px; }

.dropdown-content {
    display: none;
    position: absolute;
    background-color: yellow;}

.dropdown-content a {
    padding: 12px 16px;
    display: block;}

li a:hover {background-color: #111;}

.droplink:hover .dropdown-content {display: block;}

.dropdown-content a:hover {background-color: violet}

</style></head><body>
<ul>
<li>
<div class="droplink" href="#">Menu1 <div class="dropdown-content">
    <a href="#">Link 1</a>    <a href="#">Link 2</a>
</div></div>
</li><li>
<div class="droplink" href="#">Menu2 <div class="dropdown-content">
    <a href="#">Link 3</a>    <a href="#">Link 4</a>
</div></div>
</li><li>
<div class="droplink" href="#">Menu3 <div class="dropdown-content">
    <a href="#">Link 5</a>    <a href="#">Link 6</a>
    </div>
    </div>
</li>
</ul>
</body></html>
```

Background

Property		Possible Values	Examples
background-attachment		<i>fixed, scroll</i>	div{background-attachment:fixed;} div{background-attachment:scroll;}
background-color		valid color / color code	div{background-color:green;} div{color:orange;}
background-image		URL values.	div{background-image:url(images/img.jpg);} body {background-image:url(1.jpg);}
background-position		<i>top left, top center, top right center left, center center, center right bottom left, bottom center, bottom right</i>	div{background-position:10px 50px;} div{background-position:bottom right;}
background-repeat		<i>repeat, repeat-x, repeat-y, no-repeat</i>	div{background-repeat:repeat-x;} div{background-repeat:no-repeat;}
background	<i>background-color, background-image, background-repeat, background-attachment, background-position</i>		div{background:green url(image.jpg) no-repeat fixed center center;} div{background:url(image.jpg) fixed;}

Border

Property		Possible Values	Examples
border-top-color		valid color / color code	div{border-top-color:green;}
border-top-style		<i>none, hidden, dotted, dashed, solid, double, groove,ridge, inset, outset</i>	div{border-top-style:solid;}
border-top-width		As following predefined values: <i>Thin, medium, thick</i>	div{border-top-width:2px;} div{border-top-width:thin;}
border-top		in the following order: <i>border-top-width, border-top-style, border-top-color</i>	div{border-top:2px solid green;} div{border-top: thick double red;}
border-right-color		valid color / color code	div{border-right-color:green;}
border-right-style		<i>none, hidden, dotted, dashed, solid, double, groove, ridge, inset, outset</i>	div{border-right-style:solid;}
border-right-width		<i>thin, medium, thick</i>	div{border-right-width:2px;} div{border-right-width:thin;}
border-right		in the following order: <i>border-right-width, border-right-style, border-right-color</i>	div{border-right:2px solid green;} div{border-right:thick double red;}
border-bottom-color		valid color / color code	div{border-bottom-color:green;} div{border-bottom-color:orange;}
border-bottom-style		<i>none, hidden, dotted, dashed, solid, double, groove,ridge, inset, outset</i>	div{border-bottom-style:solid;} div{border-bottom-style:inset;}

border-bottom-width	<i>thin, medium, thick</i>	div{border-bottom-width:2px;} div{border-bottom-width:thin;}
border-bottom	<i>border-bottom-width, border-bottom-style, border-bottom-color</i>	div{border-bottom:2px solid red;} div{border-bottom:thick double red;}
border-left-color	valid color / color code	div{border-left-color:green;}
border-left-style	<i>none, hidden, dotted, dashed, solid, double, groove,ridge, inset, outset</i>	div{border-left-style:solid;}
border-left-width	<i>thin, medium, thick</i>	div{border-left-width:2px;} div{border-left-width:thin;}
border-left	in the following order: <i>border-left-width, border-left-style, border-left-color</i>	div{border-left:2px solid green;} div{border-left:thick double red;}
border-color	valid color / color code in the following order: <i>border-top-color, border-right-color, border-bottom-color, border-left-color</i>	div{border-color:green red blue olive;} div{border-color:green;} div{border-color:green red;} div{border-color:green red blue;}
border-style	<i>none, hidden, dotted, dashed, solid, double, groove,ridge, inset, outset</i>	div{border-style:solid dotted dashed double;} div{border-style:solid;} div{border-style:solid dotted;} div{border-style:solid dotted dashed;}
border-width	<i>thin, medium, thick</i>	div{border-width:1px 3px 5px 2px;} div{border-width:thin;} div{border-width:2px 4px;} div{border-width:2px 4px 5px;}
border	<i>border-width, border-style, border-color</i>	div{border:1px double green;} div{border:thin solid red;}

Margin

Property	Possible Values	Examples
margin-top margin-right margin-bottom margin-left	Lengths, percentages, & the values.	div{margin-top:5px;} div{margin-top:15%;}
margin	in the following order: <i>margin-top, margin-right, margin-bottom, margin-left</i>	div{margin:5px;} div{margin:5px 10px;} div{margin:5px 7px 4px;} div{margin:5px 12px 4px 7px;}

Dimension

Property	Examples
height, max-height, min-height, width, max-width	div{height:200px;} / div{max-width:75%;}

Classification and Positioning

Property	Possible Values	Examples
clear	<i>left, right, both, none</i>	<code>div{clear:right;}</code>
cursor	<i>url or auto,,default,pointer,move,text,wait, e-resize,ne-resize,nw-resize,n-resize,se-resize,sw-resize,s-resize,w-resize</i>	<code>div{cursor:crosshair;}</code> <code>div{cursor:url(image.csr);}</code>
display	<i>none, inline, block, list-item, run-in, compact, marker, table, inline-table, table-row-group, table-header-group, table-footer-group, table-row, table-column-group, table-column, table-cell, table-caption</i>	<code>div{display:none;}</code> <code>div{display:inline;}</code> <code>div{display:marker;}</code>
float	<i>left, right, none</i>	<code>div{float:left;}</code>
visibility	<i>isible, hidden, collapse</i>	<code>div{visibility:visible;}</code>
top, right, bottom, left	length values or percentages	<code>div{top:15px;} / div{top:2%;}</code>
position	<i>static, relative, absolute, fixed</i>	<code>div{position:absolute;}</code>
clip	Shapes, or the predefined value auto. <i>rect(top, right, bottom, left)</i>	<code>div{clip:auto;}</code> <code>div{clip:rect(2px, 4px, 7px, 5px);}</code>
overflow	<i>visible, hidden, scroll, auto</i>	<code>div{overflow:hidden;}</code>
vertical-align	Lengths, percentages, and the following predefined values: <i>baseline, sub, super, top, text-top, middle, bottom, text-bottom</i>	<code>span{vertical-align:middle;}</code> <code>td{vertical-align:top;}</code>
z-index	Integer values and the predefined value auto.	<code>div{z-index:2;}</code> <code>div{z-index:auto;}</code>

Padding

Property & Possible Values	Examples
padding-top, padding-right, padding-bottom, padding-left	<code>div{padding-top:5px;}</code> <code>div{padding-top:15%;}</code>
padding: in the following order: <i>padding-top, padding-right, padding-bottom, padding-left</i>	<code>div{padding:5px 12px 4px 7px;}</code> <code>div{padding:5px;}</code> <code>div{padding:5px 10px;}</code> <code>div{padding:5px 7px 4px;}</code>

Font

Property	Possible Values	Examples
font-family	Valid font names. Several fonts can be mentioned if one or more are unavailable	div{font-family:Arial, Helvetica, sans-serif;}
font-size	Lengths (number and unit type— i.e. <i>1em</i> , <i>12pt</i> , <i>10px</i> , <i>80%</i>) or one of the following predefined values: <i>xx-small</i> <i>x-small</i> , <i>small</i> , <i>medium</i> , <i>large</i> , <i>x-large</i> , <i>xx-large</i> , <i>smaller</i> , <i>larger</i>	div{font-size:70%;} div{font-size:0.85em;} div{font-size:medium;}
font-size-adjust	Numeric value	div{font-size-adjust:0.54;} div{font-size-adjust:0.46;}
font-stretch	normal, wider, narrower, ultra-condensed, extra-condensed, condensed, semi-condensed, semi-expanded, expanded, extra-expanded, ultra-expanded	div{font-stretch:narrower;} div{font-stretch:ultra-expanded;}
font-style	<i>normal</i> , <i>italic</i> , <i>oblique</i>	div{font-style:italic;}
font-variant	<i>normal</i> , <i>small-caps</i>	div{font-variant:normal;}
font-weight	<i>normal</i> , <i>bold</i> , <i>bolder</i> , <i>lighter</i> , <i>100</i> , <i>200</i> , <i>300</i> , <i>400</i> , <i>500</i> , <i>600</i> , <i>700</i> , <i>800</i> , <i>900</i>	div{font-weight:bolder;} div{font-weight:200;}
font	<i>font-style</i> , <i>font-variant</i> , <i>font-weight</i> , <i>font-size</i> , <i>font-family</i>	div{font:italic small-caps bold 1em 1.2em Arial } div{font:bold 0.8em Arial }

List

Property	Possible Values	Examples
list-style-type	<i>disc</i> , <i>circle</i> , <i>square</i> , <i>decimal</i> , <i>decimal-leading-zero</i> , <i>lower-roman</i> , <i>upper-roman</i> , <i>lower-alpha</i> , <i>upper-alpha</i> , <i>lower-greek</i> , <i>lower-latin</i> , <i>hebrew</i> , <i>upper-latin</i> , <i>armenian</i> , <i>georgian</i> , <i>hiragana</i> , <i>katakana</i> , <i>cjk-ideographic</i> , <i>hiragana-iroha</i> , <i>katakana-iroha</i>	ol{list-style-type:upper-roman;} ul{list-style-type:square;}
list-style-position	<i>inside</i> , <i>outside</i>	ol{list-style-position:inside;}
list-style-image	URL values.	ul{list-style-image:url(image.jpg);}
list-style	<i>list-style-type</i> <i>list-style-position</i> <i>list-style-image</i>	ul{list-style:disc inside url(image.gif);} ol{list-style:upper-roman outside;}
marker-offset	Lengths and the predefined value auto.	li:before{display:marker; marker-offset:5px;}

Table

Property	Possible Values	Examples
border-collapse	<i>collapse</i> , <i>separate</i>	table{border-collapse:collapse;}
border-spacing	Spcing	table{border-spacing:5px;} table{border-spacing:5px 10px;}
caption-side	<i>top</i> , <i>bottom</i> , <i>left</i> , <i>right</i>	caption{caption-side:top;}
table-layout	<i>auto</i> , <i>fixed</i>	table{table-layout:auto;}

Text

Property	Possible Values	Examples
color	valid color / color code	div{color:green;}
direction	<i>ltr, rtl</i> ltr = left-to-right <i>and</i> rtl = right-to-left	div{direction:ltr;} div{direction:rtl;}
line-height	predefined value of <i>normal</i>	div{line-height:normal;} div{line-height:2em;} div{line-height:125%;}
letter-spacing	predefined value of <i>normal</i>	div{letter-spacing:normal;} div{letter-spacing:5px;}
text-align	<i>left, right, center, justify</i>	div{text-align:center;} div{text-align:right;} td{text-align:"."};
text-decoration	<i>none, underline, overline, line-through, blink</i>	div{text-decoration:none;}
text-indent	Lengths and percentages.	div{text-indent:12px;} div{text-indent:2%;}
text-shadow	A list contains a color followed by numeric values (separated by spaces) that specify: 1. The color for the shadow effect 2. Horizontal distance to right of text 3. Vertical distance below the text 4. Blur radius	div{text-shadow:green 2px 2px 7px;} div{text-shadow:olive -3px -4px 5px;}
text-transform	<i>none, capitalize, uppercase, lowercase</i>	div{text-transform:uppercase;}
white-space	<i>normal, pre, nowrap</i>	div{white-space:pre;}
word-spacing	A length (in addition to the default space) or the predefined value of <i>normal</i> .	div{word-spacing:normal;} div{word-spacing:1.5em;}

Outline

Property	Possible Values	Examples
outline-color	valid color / color code	div{outline-color:green;}
outline-style	<i>none, dotted, dashed, solid, double, groove, ridge, inset, outset</i>	div{outline-style:solid;}
outline-width	Lengths or the following predefined values: <i>thin, medium, thick</i>	div{outline-width:2px;} div{outline-width:thin;}
outline	Separate values by a space in the following order <i>outline-color, outline-style, outline-width</i>	div{outline:green solid 2px;} div{outline:orange double thick;}

span tag

The HTML `` tag is used for grouping and applying styles to inline elements. There is a difference between the span tag and the div tag. The span tag is used with inline elements whilst the div tag is used with block-level content. However this tag supports any attribute as similar to 'div' tag.

`<p>It is a paragraph It is a wordIt is a paragraph</p>`